Army Research Laboratory

Predicting Morphology of Polymers Using Mesotek+

by Tanya L. Chantawansri, Erin M. Lennon, and Jan Andzelm

ARL-TR-5087 February 2010

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5069

ARL-TR-5087 **February 2010**

Predicting Morphology of Polymers Using Mesotek+

Tanya L. Chantawansri
Oak Ridge Institute for Science and Education

Erin M. Lennon
Northwestern University

Jan Andzelm
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* February 2010	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** June 2009–October 2009

4. TITLE AND SUBTITLE Predicting Morphology of Polymers Using Mesotek+	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Tanya L. Chantawansri,[*] Erin M. Lennon,[†] and Jan Andzelm	**5d. PROJECT NUMBER** H84 **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-WMM-G Aberdeen Proving Ground, MD 21005-5069	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-5087
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

[*]Oak Ridge Institute for Science and Education, 4692 Millennium Dr., Ste. 101, Belcamp, MD 21017
[†]Northwestern University, Department of Engineering Science and Applied Mathematics, 2145 Sheridan Rd., Evanston, IL 60208

14. ABSTRACT

When new polymeric materials are designed, it is desirable if computational modeling can be used to predict microscale self-assembly and physical properties. Field theoretic methods such as self-consistent field theory have been used to accurately model the self-assembly behavior for a variety of dense polymeric systems of high molecular weight. This theory, as developed by Glenn Fredrickson and coworkers, has been programmed into a C++ code called Mesotek+. This report presents a theory overview and an introduction on how to use the program. An input file is then produced for Mesotek+ to reproduce the phase behavior for an experimental system of poly (styrene-b-isoprene) in the solvent tetradecane (C-14).

15. SUBJECT TERMS

polymer, morphology, Mesotek

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Tanya L. Chantawansri
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	30	**19b. TELEPHONE NUMBER** (*Include area code*) 410-306-4008

Contents

List of Figures

List of Tables

Acknowledgments

We would like to thank Professors Glenn H. Fredrickson and Eric Cochran for providing the Mesotek+ program.

This research was supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Research Laboratory (ARL) administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and ARL.

1. Overview

Mesotek+ is a parallel C++ mesoscale polymer modeling program that uses the field-theoretic formalism developed by Glenn Fredrickson and coworkers at the University of California, Santa Barbara. This code is generalized for a variety of non-closed-loop polymer architectures such as linear, star, and branched, and it can be applied to model the equilibrium morphology of both neat polymeric systems and nanocomposites. Two different field theories, self-consistent field theory and complex Langevin theory, are implemented. Self-consistent field theory is appropriate when thermal fluctuations can be ignored, such as in dense, high-molecular weight polymer melts. Complex Langevin theory should be used for polymer solutions in the dilute or semidilute regimes, microemulsion and micellar phases, block copolymers near the order-disorder transition, and blends near a critical point, second-order phase transition, or weakly first-order phase transitions (*1*). Although complex Langevin theory can also be used to look at dense polymer melts, it is much more computationally expensive than self-consistent field theory and should be used only when necessary.

When the neat polymeric systems are modeled, two routes can be taken to reach equilibrium: diffusive dynamics or optimization. Although these two methods transverse different pathways, they should both reach the same equilibrium state. Diffusive dynamics can be used if the kinetic pathway toward equilibrium is important; however, the results should be analyzed with caution since it does not include hydrodynamic and elastic stress couplings, which are often important (*1*). The optimization pathway occurs through fictitious dynamics that are not physically realistic. If only the equilibrium morphology is desired, optimization should be used because it allows for a quicker route to equilibrium than diffusive dynamics.

Nanocomposites can only be modeled using the optimization route with self-consistent field theory. In this method, the polymers are modeled using self-consistent theory for fixed nanoparticles, while the nanoparticles are moved using sampling methods such as Monte Carlo or Langevin (*2*).

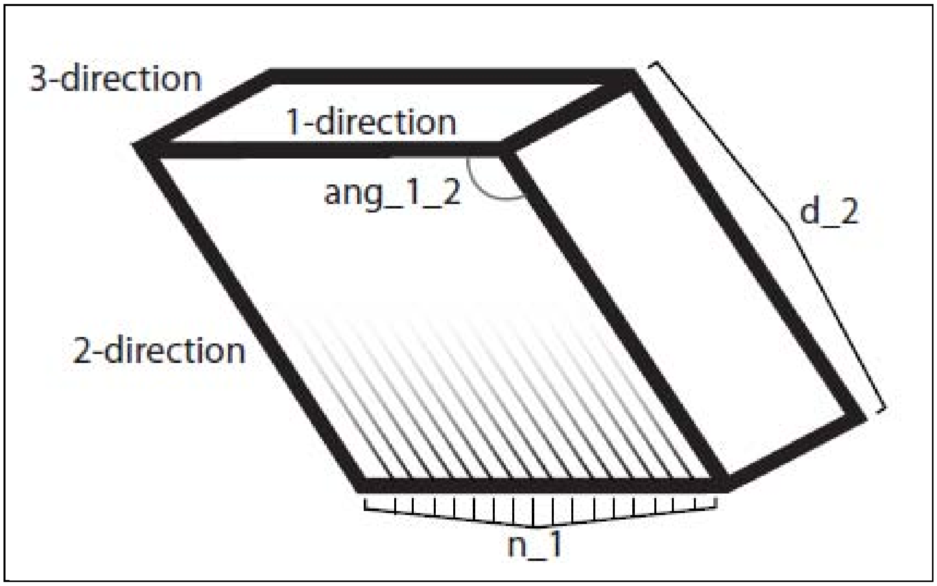
This report presents an introduction on how to use Mesotek+ (it is assumed that the reader has a basic knowledge of field-theoretic methods). If the reader is interested in the underlying theory used in this software, a variety of articles are available (*1*, *3*).

2. Defining Your System

To run a simulation, it is necessary to produce an input file that includes your system's parameters. The minimum information needed is the simulation box size (discretization and dimensions), details regarding your species (type and interaction), and components (architecture and fractions).

2.1 Box Size

To parameterize your simulation box, it is necessary to specify the box angles, the resolution of each dimension, and the length of the simulation box (see figure 1). This group is rather straightforward. Here, the rank is the dimension of your system. In a one-dimensional simulation, the dimensions not specified are assumed to be homogeneous such that only the lamellar or disordered phases are observed. For a two-dimensional (2-D) simulation, the third direction that is not explicitly simulated is assumed to be uniform such that three-dimensional (3-D) spheres will appear as cylinders in 2-D. The simulation box is assumed be periodic unless a confined geometry is specified. The parameters n_i and d_i are the number of collocation points and lattice length, respectively, in the i direction (i = 1, 2, or 3). If complex Langevin is used, it is also necessary to include the line fftw_mode = c2c, which specifies the use of complex fields.



Note: Box size (rank = 3; n_1 = 32; n_2 = 64; n_3 = 16; d_1 = 8.0; d_2 = 12.0; d_3 = 4.0; ang_1_2 = 120; ang_1_3 = 90; ang_2_3 = 90).

Figure 1. Simulation box that includes the parameters defined in the box size group.

2.2 Species

The species indicate the species in your system as well as the interactions. Mesotek+ has two different types of inputs: experimental and theoretical.

2.2.1 Theoretical

When theoretical units are chosen, all length scales are scaled by the polymer's radius of gyration (R_g), and the statistical segment length is assumed to be equal for each species. A subgroup must be produced for each species as well as for all interactions. For example, for an AB homopolymer blend or any other polymer architecture with two polymer species (e.g., AB, ABA, etc.), you must make a subgroup for A and a subgroup for B, as well as a sub-subgroup for the interaction chi_AB. If diffusive dynamics are used instead of optimization, the diffusion coefficient must also be specified through the parameter diffusivity. The sample code for this model is in code 1a.

Code 1a: Sample Code for Theoretical

```
Parameters
Species{
  A{
     Rg2/M = 1.;
     mass density= 1.;
     type = Gaussian;
     diffusivity = 1;
    }
   B{
     Rg2/M = 1.;
     mass density= 1.;
     type = Gaussian;
     diffusivity = 1;
}
    chi {
       A_B{
            A = 0,2;
       }
   }
}
```

Code 1b: Sample Code for Experimental

```
Species{
  A{
     Rg2/M = 1.;
     mass density= 1.;
     type = Gaussian;
    }
   B{
     Rg2/M = 1.;
     mass density= 1.;
     type = Gaussian;
}
    C{
     Rg2/M =1.;
     mass density = 1.;
     type = Solvent;
}
    chi {
       A_B{
            A = 0.2;
       }
       B_C{
            A = 0.1;
       }
       A_C{
            A = 0.2;
        }
   }
}
```

If an additional species is added, such as a solvent, an additional species group C would have to be defined along with two additional sub-subgroups for the interactions chi_AC and chi_BC (see code 1b). In this example, the interaction between the different species is given by the variable A, which is the minimum constant required to define χ, the Flory-Huggins interaction parameter.

2.2.2 Experimental

When experimental units are chosen, the length scales are in physical dimensions (angstroms). The code structure for species is similar to the structure used in the theoretical system but will include additional information about the experimental system. When the polymers are described, it is necessary to input two out of three values per species (mass density, molecular weight of a bead [M0], or molar volume of a bead) as well as one experimental length scale (either the end-to-end distance squared per molecular weight [R0^2/M] or the radius of gyration squared per molecular weight [Rg2/M]). The end-to-end distance is related to the radius of gyration by a simple constant (R0^2 = 6 Rg2). To calculate these two values, which are used to describe the dimensions of a polymer chain, it is necessary to calculate the Kuhn length. The Kuhn length (b) is a measure of the conformational rigidity and is the product of the contour length over the repeating structural unit (l_o) and the characteristic ratio (C_∞). Here, the characteristic ratio is a measure of short-range interactions along with the contour length over the repeating structural unit. This parameter can be calculated through molecular modeling or obtained through a polymer handbook. The radius of gyration or the end-to-end distance can then be calculated using Rg2 = b^2 N_b /6 or R0^2 = b^2 N_b, respectively, where N_b is the number of beads representing our polymer chain.

There are several ways to calculate the Flory-Huggins interaction parameter, χ, in this system. One way is to use a temperature and composition-dependent interaction parameter: χ_{ij} = A + B/T + C*density$_i$(r). If this form of the interaction parameter is used, a minimum of the A parameter must be defined. The values of some polymer-polymer interactions can be found in polymer handbooks. Another way to define the interactions is through solubilities: χ_{ij} = V0(δ_i – δ_j)^2 /RT. These solubility parameters (δ) can be found in polymer handbooks as well as from molecular dynamics simulations. In both of these methods, it is proper to define V0, which is a reference volume in units of cm^3/mol, and the temperature in kelvins. If V0 is not defined, it is set to the default value of 71.1 cm^3/mol. Interactions between species are scaled by N, the index of polymerization, since the interaction increases with the size of the polymer. Values of χN calculated through these two methods tend to be very large, which may lead to numerical instabilities. If this is the case, the interactions should be capped to a maximum value. Again, diffusive dynamics can also be used in the same way as in the theoretical case. Two different example codes for an experimental system composed of a styrene polymer and a benzene solvent are shown in codes 2a and 2b. See section 6 for an example of how to calculate some of these experimental values.

Code 2a: Sample Code for Theoretical

```
species
{
  V0 = 70.0;
  temp = 500;
  Styrene
  {
   type = gaussian;
    R0 ^ 2 / M = 0.434;
    mass density = 0.969;
    solubility parameter = 19;
  }
  Benzene
  {
    type = solvent;
    mass density = 0.874;
    M0 = 78;
    solubility parameter = 18.8;
  }
  chi
   {
    max chi*<N> = 90;
    Styrene_Benzene
       {
          solubility parameter = true;
       }
    }
}
```

Code 2b: Sample Code for Experimental

```
species
{
  V0 = 70.0;
  temp = 500;
  Styrene
  {
    Rg ^ 2 / M = 0.134;
    mass density = 0.969;
    type = gaussian;
   }
  Benzene
  {
    type = solvent;
    mass density = 0.874;
    M0 = 78;
  }
    chi
     {
        max chi*<N> = 90;
         Styrene_Benzene
           {
              A = 0.2;
              B = 5.0;
            }
        }
}
```

2.3 Components

The Component tab is used to define the architecture of the system. The basic structure of this group is as follows:

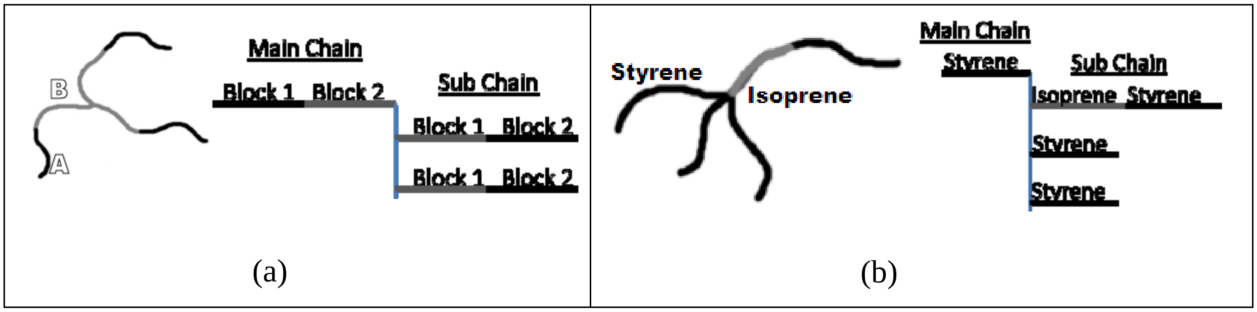
```
components{
        Ns_per_N0 = 100;             \\ resolution along the polymer chain
        firstcomponentname{
                 <component properties>
                 blocks{
                        block 1{
                              < block 1 properties>
                         }
                         … more blocks …
                  }
                 subchains{
                       subchain 1 {
```

```
                    < subchain 1 properties>
                    …     blocks    ….
                    …    subchains …
                  }
                 … more subchains …
                }
               …  more components … {
               }
           }
}
```

Here, the blocks and subchains groups are used to define the architecture of the polymer, and the block subgroup within the blocks group is used to define which polymers are present when one transverses down the polymer backbone. The subchain group adds chains to the end of the last polymer block, where the subchain subgroup can be used to add additional blocks. Again, there are theoretical and experimental formats for this group.

2.3.1 Theoretical and Experimental Variables

For the theoretical variables, we will look at an AB three-arm diblock copolymer star polymer, where each arm is a symmetric diblock copolymer (see figure 2a) in a C solvent. The inner star is composed of the B block. For the experimental variable, we look at a styrene-isoprene four-arm star polymer. In this example, one arm is a styrene-isoprene diblock copolymer where the isoprene end is attached to three styrene arms (see figure 2b).



(a)

(b)

Figure 2. Star polymer associated with (a) theoretical code 3a and (b) experimental code 3b.

2.3.2 Theoretical and Experimental Setup

In the theoretical setup, the volume fraction (f) for each species is specified to be 0.5 relative to each chain, where the index of polymerization (N) is specified to be 100 for each arm. In this gel, the volume fraction of the star is 0.9 and the solvent is 0.1. In the experimental setup, the volume fraction f is ignored, and instead, the total molecular weight of the polymer is used along with the mole fraction of each block relative to the entire star polymer. In this setup, only the molecular weight of the first block of the main chain is read in (Mn = Total Mn * mole fraction [first block]), and the molecular weight of the additional blocks are calculated internally. In the experimental input, it is not necessary to calculate N; instead, the program will calculate Nusing Mn. In the experimental setup, it is also possible to input the molecular weight of each block and the number of beads that represents each block (Ns) instead of the number of beads used to represent the entire polymer (Ns_per_N0). An example code for the theoretical and experimental system can be found in codes 3a and 3b, respectively.

Code 3a: Sample Code for Theoretical

```
components {
      Ns_per_N0 = 100;
      star {
           type = polymer;
           vol_frac = 0.9;
           N = 100;
           blocks {
                   block 1 {
                        species = A;
                        f = 0.5;
                    }
                    block 2 {
                         species = B;
                         f=0.5;
                     }
              }
              subchains{
                   subchain 1{
                   N = 100;
                   blocks{
                        block 1 {
                           species = B;
                           f = 0.5;
                       }
                         block 2 {
                            species = A;
                             f = 0.5;
                       }
                    }
                   subchain 2{
                   N = 100;
                   blocks{
                        block 1 {
                           species = B;
                           f = 0.5;
                       }
                         block 2 {
```

Code 3b: Sample Code for Experimental

```
components{
  Ns_per_N0 = 30;
  diblock{
    type = polymer;
    vol_frac = 1.0;
    blocks{
      styrene block{
        species = Styrene;
        mole fraction = 0.2;
          Mn = 40000;
      }
    }
    subchains{
      styrene subchain{
        blocks{
          A{
            species = Styrene;
            mole fraction = 0.2;
           }
         }
      }
      isoprene subchain{
        blocks{
          A{
            species = Isoprene;
            mole fraction = 0.2;
               }
           }
        subchains{
          styrene subchain 1{
            blocks{
              A{
                species = Styrene;
                mole fraction = 0.2;
                  }
              }
          }
```

```
              species = A;
              f = 0.5;
           }
          }
        }
      }
    }
  solvent{
     type = solvent;
     vol_frac = 0.1;
     species = C;
  }
}
```

```
     styrene subchain 2{
      blocks{
       A{
        species = Styrene;
        mole fraction = 0.2;
       }
      }
     }
    }
   }
  }
 }
}
```

3. Defining Simulation Methods and Parameters

Now that the polymer system is defined, we will go over some simulation methods, such as numerical methods, format and intervals of output files, initial conditions, and iteration control.

3.1 Initial Field Condition

This is typically the first line in the input file and it specifies the initial format of the field. When the initial guess property is set as initial_guess = random. The fields are produced using a random number generator. For debugging, it is easier to use a constant zero initial condition, which can be specified by setting initial_guess = zero. If field configurations are appended to the end of the input file, they will be read in by setting initial_guess = attached, or if the initial condition is written to a separate file, the initial guess is set to the file name. For example, if the fields are saved in a file named fields.txt, they can be read in by specifying initial_guess = fields.txt. It is also possible to use spectral seeding to guess the symmetry of the system. This will be covered later in this report.

3.2 Iteration Control

This group specifies the maximum number of iterations, max_iter, that will be conducted if the minimum error, min_error, is not reached. If the minimum error is reached before the maximum iteration, the simulation will finish. Within this subgroup it is also possible to ramp multiple parameters such as χ, angles, or simulation box size length after a specified iteration count is met. It is also optional to specify how the parameter is sampled, either linear or log. If it is not specified, linear sampling is assumed. The iteration control group is defined as follows for a maximum of 1000 iterations and a minimum error of 1e-5 for seven box sizes. The x box length (d_1) is varied from 8 to 14 (abs_range), by adding 1 to the length 6 times (n_step).

```
iteration control {
     max_inter = 1000;
     min_error = 1.e-5;
     vary {
        1 {
              parameter = d_1;
               abs_range = 8.,14.;
               n_step = 6;
               first_step = 1;
               }
        }
}
```

3.3 Field Relaxation

To update the fields in self-consistent field theory through optimization or diffusive dynamics and in complex Langevin theory, it is necessary to specify desired numerical schemes and the time steps.

For diffusive dynamics, the scheme should be set to diffusive dynamics, and the diffusive time step should be specified. Here, the fields are updated every iteration (n_iter), starting from the first iteration (first_iter). Here, the sis_const is the relaxation constant for the pressure field update.

```
field relaxation
{
  first_iter = 0;
  n_iter = 1;
  scheme = diffusive dynamics;
  sis_const = 5.0
  diffusive time step=1;
}
```

When self-consistent field theory optimization or complex Langevin is used, there are two methods that can be used: the explicit Euler method or the semi-implicit-seidel method.

The first method is the explicit Euler method. When scft optimization is used, this group takes the following form where euler_const_1 is the relaxation constant for the pressure field and euler_const_2 is the relaxation constant for the nonpressure field.

```
field relaxation
{
  first_iter = 0;
  n_iter = 1;
  scheme = euler;
  euler_const_1 = 1;
  euler_const_2 = 0.1;
}
```

If complex Langevin is instead desired, it is necessary to add two additional parameters to the group: complex langevin = true and ginzburg = 100.

Another more stable method to update the fields is the semi-implicit-seidel method developed by Ceniceros and Fredrickson (*4*). Here sis_const is the field relaxation constant for the pressure field, while sis_const_minus is the relaxation constant for the nonpressure field. It is also possible to allow the value of the relaxation constants to be rescaled through the simulation by including auto lambda = true and auto lambda restart = true (can also be used in diffusive dynamics and in the Euler method).

```
field relaxation
{
  first_iter = 0;
  n_iter = 1;
  scheme = sis;
  sis_const = 5.;
  sis_const_minus = 1.;
  auto lambda = true;
  auto lambda restart = true;
}
```

3.4 Status Save

If there is a parameter value that you would like calculated every n_iter iterations, such as the free energy, box size, and angles, the Status Save group can be used to store the output. Each variable that you would like to save must be on a new line followed by the variable type and the type of output.

In the following example, the code will save data to a file at every iteration and will dump the iteration number (the box size in the 1-direction) and the χ_{AC} into a file.

```
status save {
      n_iter = 1;
      outputs = restart file-> iteration, int, %05d
            results->F, double, %24.16g
            box size->d_1, double,%8.3f;
            species-> chi->A_C->A,double,%4.2f
}
```

3.5 Field Save

When running your job, it is sometimes important to restart your job. In order to do this, it is necessary to save your fields throughout the simulation. The Field Save group can be used to save the fields at a specified frequency or after the error change is greater than a given threshold.

In addition to the restart capability, the data saved using this group can also be used to visualize results before the simulation is completed.

In the following example, the fields are saved in text format every 100 iterations or if the error change is >0.4, but not after each iteration.

```
field save{
    n_iter = 100;
    format = text;
    save recent = false;
    delta error = 0.4;
}
```

For restarting the run, the data should be saved in text format, but for visualization, two additional formats can be saved: gnuplot and binary.

3.6 Misc

This group dictates the numerical methods used to solve the modified diffusion equation for the propagators and the quadrature used to calculate the densities. In addition, the verbosity of the simulation is indicated through this group. The verbose level indicates how much information is sent to the terminal or to the standard output, where a level of 0 will produce no output and a level of 6 will print out the value of every array at every iteration. High verbose levels are used for debugging purposes.

Two methods used to calculate the modified diffusion equation are the operator splitting method (op_split) or the backward differentiation formula (bdf) (*5*). Two quadratures can be used, Gaussian (gauss) or Simpson (simpson).

```
misc{
    verbose level = 2;
    quadrature = gauss;
    diffusion equation {
            algorithm = op_split;
    }
}
```

4. Extensions

4.1 Unit Cell Calculations

To calculate an optimal unit cell of your morphology, you can use the box relaxation group, which will resize the simulation box length and/or box angles to an optimal cell with minimum stress.

In the following example, the box is resized every iteration after the fifth iteration as long as the error is <1.e-3 and the total stress on the box is >1.e-4. The length d_1 is relaxed at a rate constant of 0.1, where d_2 is constrained to the same length. The third length d_3 and all the angles are not allowed to change.

```
box relaxation{
  first_iter = 5;
  n_iter = 1;
  min_error = 1.e-3;
  min_stress = 1.e-4;
  speed = 0.1;
  which_dims{
        d_1 = true;
        d_2 = constrain(d_1);
        d_3 = false;
        ang_1_1 = false;
        ang_1_2 = false;
        ang_2_3 = false;
        d_1 bounds = 10,15;
        d_2 bounds = 10,15;
        }
}
```

4.2 Spectral Seeding

It is possible to seed in simple symmetries into the simulation. One can use centrosymmetric cosine seeding (-id) or plane wave basis seeding (id).

```
spectral{
   space group = -id;
   weights = 4, 1;
}
```

The weights indicate a prefactor for each harmonic. For this case, only the first two harmonics are used: 4 cos(2πx/D) + cos(4πx/D). If more weights are included, more harmonics are used.

4.3 Ensemble of Particles

To implement particles, we can use the ensemble group, where the nanoparticles will be moved a total of specified moves (total_moves). The particles will be moved after the error tolerance specified in the Iteration Control group is reached. In setting up the nanoparticles, a specified number of position updates will be performed between additions of particles (moves_per_step). The ratio of the variable delta over the variable ginzburg in the Field Relaxation group defines the ratio between the force-biased term and the random term in the position update, where delta is the step width that specifies the maximum distance each particle can move at a given step. The method variable indicates the method that will be used to move the particles. There are three choices: smart Monte Carlo (SMC), Langevin dynamics (langevin), and standard Metropolis Monte Carlo (MC). The three methods are performed slightly differently. In SMC, the particles are displaced based on the force and a random force based on the step width. Langevin displaces particles using the same method as SMC where all moves are accepted unless it leads to overlap, but the step size must be small for the calculation to be accurate and stable. MC displaces particles based on a random force calculated from the step width, but it could lead to artificial clustering of particles. The maximum distance a particle can move is specified using the cutoff variable. The thickness and overlap_margin variables specify the width of the interfacial region of the particle cavity function and the margin used to prevent particles from overlapping, respectively.

In the following example, the 4 particles of radius 75 are moved 100 times where they are initially placed at random with one position step between particle additions. The particles are allowed to move a maximum step width of 125 using the SMC algorithm. Thickness and overlap margins of 25 are used.

```
ensembles
{
  total_moves = 100;
  moves_per_step = 1;
  delta = 125;
  cutoff = 1000;
  method = SMC;
  setup_condition = random;
  rotations_per_move = 0;
  max_rotation = 0.0;
  freq_reinsert = 0;
  num_reinsert = 0;
  ensemble 1
  {
    species = nano;
    radius = 75;
    thickness = 25.000;
    overlap_margin = 25.000;    total_number = 4;
    add_per_step = 4;
    type = sphere;
  }
}
```

5. Visualization

To visualize this data, it is possible to use a variety of programs. For 2-D data, gnuplot is the easiest to use since it is possible to have the program output the data in this format. For 3-D visualization, there are a variety of other programs that can be used to visualize data, such as Matlab, IFrIT (*6*), and a viewer produced by Eric Cochran specifically for this program. All three of these programs will involve saving data in the text format, but additional scripts or programs must be used to extract the field values from these files when IFrIT or Matlab are used. For these scripts, please email tanya.chantawansri@arl.army.mil.

6. Conversion of Experimental Data for Mesotek+

We will consider a poly(styrene-b-isoprene) in the solvent tetradecane (C-14). The phase diagram can be found in Bang et al. (*7*).

The experiments should provide the molecular weight/total mass of each polymer species in the block copolymer or the total molecular weight and mole/weight/mass fractions.

In this report, we are given the mass of styrene and isoprene in the diblock copolymer:

Diblock: mass of styrene = 15,200 g/mol and mass of isoprene = 15,400 g/mol.

From these species, we can calculate the physical properties of the monomers (see table 1). These properties were obtained via the program Synthia (*8*) of Material Studio, but they can also be found in polymer handbooks. For the solvent, the repeat unit and characteristic ratio are not required.

Table 1. Properties for isoprene, styrene, and C-14 obtained using Synthia.

Species	**Monomer Volume (cm^3/mol)**	**Monomer Mass (g/mol)**	**Repeat Unit Length (Å)**	**Characteristic Ratio**	**Solubility Parameter**
Isoprene	76.6	68.1	4.4	5.4	16.9
Styrene	97.0	104.2	2.5	9.9	20.1
C-14	225.1	196.3	NA	NA	17.5

Note: NA = not applicable.

First it is necessary to calculate the volume fraction of isoprene and styrene. This can be done using the density of each component: styrene density = 1.047 g/cm^3 and isoprene density = 0.913 g/cm^3.

The volume of styrene is calculated by dividing the mass of styrene in the diblock by the density of styrene: (15,200 g/mol)/(1.047 g/cm^3) = 14517.67 cm^3/mol.

The volume of isoprene is calculated by dividing the mass of isoprene in the diblock by the density of isoprene: (15,400 g/mol)/(0.913 g/cm^3) = 16,867.5 cm^3/mol.

The volume fraction of styrene (F_S) and isoprene (F_I) in the diblock can then be calculated by dividing the volume of the species of interest by the total volume:

F_S = 14517.67 cm^3/mol / (14517.67 cm^3/mol + 16867.5 cm^3/mol) = 0.46.

F_I = 16867.5 cm^3/mol / (14517.67 cm^3/mol + 16867.5 cm^3/mol) = 0.54.

Using these values, we can calculate the volume and molecular weight of a bead. Let us define our polymer as having 60 total beads, which will give us 28 styrene beads and 32 isoprene beads.

Relevant equations include the total molecular weight $M_N = n_I M_I + n_S M_S$, the volume fraction of each species $F_I = n_I V_I / (n_I V_I + n_S V_S)$, and $F_S = n_S V_S / (n_I V_I + n_S V_S)$, where M_N is the total mass of a polymer chain, M_I is the mass of an isoprene monomer, M_S is the mass of a styrene monomer, F_I is the volume fraction of an isoprene monomer, F_S is the volume fraction of a styrene monomer, V_I is the volume occupied by a monomer of isoprene, and V_S is the volume occupied by a monomer of styrene.

Combining these three equations, we define a new parameter C: $C = M_N / (M_I F_I V_S + M_S F_S V_I)$. Using the values calculated through Synthia (*8*), we get the following value for C:

C = (15,200 g/mol + 15,400 g/mol) / (68.1g/mol × 0.54 × 97.0 cm^3/mol + 104.2 g/mol × 0.46 × 76.6 cm^3/mol) = 4.23 mol/cm^3.

We can then calculate the number of isoprene and styrene monomers denoted by n_S and n_I, respectively.

$n_I = C F_I V_S$ = 4.23 mol/cm^3 × 0.54 × 97.0 cm^3/mol = 221 monomers.
$n_S = C F_S V_I$ = 4.23 mol/cm^3 × 0.46 × 76.6 cm^3/mol = 149 monomers.

We then need to calculate how many monomers are in a bead for each component. For styrene, we have 149 monomers in 28 beads = 5.32. For isoprene, we have 221 monomers in 32 beads = 6.91.

To calculate the volume of each bead, we multiply the volume for a monomer by the following ratio:

Bead volume for styrene = 5.32 × 97.0 cm^3/mol = 516.04 cm^3/mol.
Bead volume for isoprene = 6.91 × 76.6 cm^3/mol = 529.31 cm^3/mol.

To get the mass of each bead, we multiply the mass of a monomer by the following ratio:

Bead mass of styrene = 5.32 × 104.2 g/mol = 554.34 g/mol.
Bead mass of isoprene = 6.91 × 68.1 g/mol = 470.57 g/mol.

To get the Kuhn length, we multiply the repeat unit length and characteristic ratio:

Kuhn length for styrene = 2.5 Å × 9.9 = 24.8 Å.
Kuhn length for isoprene = 4.4 Å × 5.4 = 23.8 Å.

The Kuhn length is related to the radius of gyration. To calculate the parameter Rg^2/M, we multiply the square of the Kuhn length and divide by the product of the bead mass and the number 6:

Rg^2/M for styrene = 24.8 Å × 24.8 Å / (6 × 554.3 g/mol) = 0.185 Å^{2}g/mol.
Rg^2/M for isoprene = 23.8 Å × 23.8 Å / (6 × 470.6 g/mol) = 0.201 Å^{2}g/mol.

Since we are representing the solvent as one bead, we can use the parameters obtained by Synthia (*8*) for our calculations. Table 2 summarizes the parameters that should be inputted into Mesotek+.

Table 2. Calculate input parameters for Mesotek+.

Species	**Bead Volume (cm^3/mol)**	**Bead Mass (g/mol)**	**Solubility**	**Rg^2/M (Å^{2}g/mol)**
Styrene	516.0	554.3	20.1	0.185
Isoprene	529.3	470.6	16.9	0.201
C14	225.1	196.3	17.5	NA

Note: NA = not applicable.

The following pages present a script for this experimental system. Here, the molecular weight of each block is off from what is given experimentally due to roundoff errors when the bead mass and volume were calculated, but this difference should not affect the results. Here, a 2-D simulation of a 40/60 poly(styrene-b-isoprene)/C14 mixture at 373 K is considered, which should correspond to the cylindrical phase. Results are shown in figure 3.

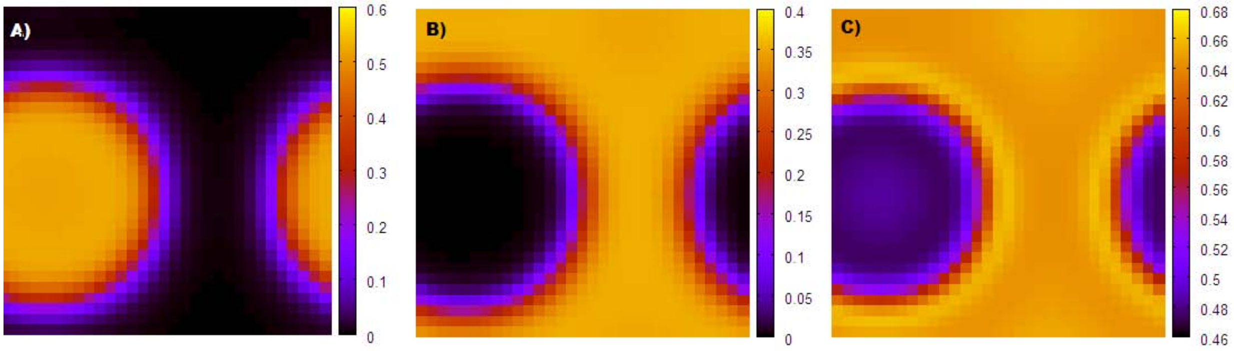


Figure 3. Results from 40/60 volume styrene-b-isoprene + tetradecane using gnuplot: A) styrene volume fraction, B) isoprene volume fraction, and C) tetradecane volume fraction. The color bar to the right of each plot indicates how the coloring corresponds to the volume fraction.

```
initial guess = random;

components
{
  polymer
  {
    type = polymer;
    vol_frac = 0.4;
  blocks
 {
   Bead Styrene
   {
    species = Styrene;
    Mn = 15520.4;            // Ns * Bead Mass ~ 152,000
    Ns = 28;
   }
   Bead Isoprene
   {
    species = Isoprene;
    Mn = 15059.2;           // Ns * Bead Mass ~ 154,000
    Ns = 32;
   }
  }
 }
 c14
 {
  type = solvent;
  vol_frac = 0.6;
  species = C14;
 }
}

species
{
 temp = 373;
 C14
 {
  type = solvent;
  solubility parameter = 17.5;
  M0 = 196.3;
  Rg2/M = 0.217219;
  molar volume = 225.1;
 }
 Styrene
 {
  type = gaussian;
  solubility parameter = 20.1;
  M0 = 554.3;
  Rg2/M = 0.18493;
  molar volume = 516;
 }
 Isoprene
 {
  type = gaussian;
```

```
    solubility parameter = 16.9;
    M0 = 470.6;
    Rg2/M = 0.201;
    molar volume = 529.3;
  }
  chi
  {
    max chi*<N> = 90;
    C14_Styrene
    {
      solubility parameter = true;
    }
    C14_Isoprene
    {
      solubility parameter = true;
    }
    Isoprene_Styrene
    {
      solubility parameter = true;
    }
  }
}

iteration control
{
  max_iter = 10000;
  min_error_plus = 0.0001;
  min_error_minus = 0.0001;
}

field relaxation
{
  first iter = 1;
  n_iter = 1;
  scheme = sis;
  plus scheme = sis;
  minus scheme = sis;
  sis_const = 5;
  sis_const_minus = 0.2;
  auto lambda = true;
}

box size
{
  rank = 2;
  n_1 = 32;
  n_2 = 32;
  d_1 = 320;
  d_2 = 320;
  ang_1_2 = 90;
}

field save
{
  n_iter = 100;
  format = gnuplot;
  delta = 50;
  save recent = 100;
}


misc
{
  quadrature = simpson;

  diffusion equation
  {
    algorithm = split;
    extra sampling = 1;
  }
}
```

7. References

1. Fredrickson, G. *The Equilibrium Theory of Inhomogeneous Polymers (International Series of Monographs on Physics)*; Oxford University Press: New York, 2006.

2. Sides, S. W.; Kim, B. J.; Kramer, E. J.; Fredrickson, G. H. Hybrid Particle-Field Simulations of Polymer Nanocomposites. *Phys. Rev. Lett.* **2006**, *96*, 250601.

3. Fredrickson, G. H.; Ganesan, V.; Drolet, F. Field-Theoretic Computer Simulation Methods for Polymers and Complex Fluids. *Macromol.* **2002**, *35* (1), 16–39.

4. Ceniceros, H. D.; Fredrickson, G. H. Numerical Solution of Polymer Self-Consistent Field Theory. *Multiscale Model. Simul.* **2004**, *2*, 452.

5. Cochran, E. W.; Garcia-Cervera, C. J.; Fredrickson, G. H. Stability of the Gyroid Phase in Diblock Copolymers at Strong Segregation. *Macromol.* **2006**, *39* (7), 2449–2451.

6. Google Sites Home Page. http://sites.google.com/site/ifrithome/ (accessed 4 January 2010).

7. Bang, J.; Viswanathan, B.; Lodge, T. P.; Park, M. J.; Char, K. Temperature-Dependent Micellar Structures in Poly(styrene-b-isoprene) Diblock Copolymer Solutions Near the Critical Micelle Temperature. *J. Chem Phys.* **2004**, *121* (22), 11489.

8. Accelrys Home Page. http://accelrys.com/products/materials-studio/modules/synthia.html (accessed 4 January 2010).

NO. OF COPIES	ORGANIZATION
1 (PDF only)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LAB IMNE ALC HRR 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIM L 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB RDRL CIM P 2800 POWDER MILL RD ADELPHI MD 20783-1197
	ABERDEEN PROVING GROUND
1	DIR USARL RDRL CIM G (BLDG 4600)

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
20	DIR USARL RDRL WMM G T CHANTAWANSRI